Learning to prototype rules of the business

how do our students get on with the

1. *course*
2. *tool, and*
3. *design formalisms*

Lex Wedemeijer

course: Rule-based Design for BPMIT

- **Business Process Management and IT**

  approx. 40 students/year

  virtually no training in design or in formal approaches

- **learning goals**

  1. understand the ideas, theory and approach of Business Rules
  2. create a small Rule-based Design (in wiki + Ampersand tool)
     - *model with 5 to 15 relations,*
     - *with multiplicities,*
     - *with some 5 composite Business Rules, and*
     - *with initial populations that show NO violations (or else)*

not required:

- *the workflow (process),*
- *user interface,*
- *editing of live data,*
- *quality check on the specs*

Student feedback: the good

- **general opinion on the course: OK**
  - judged as 'difficult' to 'very difficult'
  - "very interesting", "relevant", "captivating"
  - "I now realize the importance of being precise in requirements because the Ampersand tool sanctions even the tiniest errors"

- **it is all about the business logic**
  - "once you get the idea, then getting the rules formalized is rather straightforward"
  - "before, I had no idea what could be achieved by way of business rules and Relation Algebra"
  - "semiformal language (RuleSpeak) is really important"
  - "IF ... THEN MUST ... rule syntax is helpful"

Student feedback: the bad

- **look but don't touch**
  - fascinating theory
  - overwhelming and confusing
  - "every time I thought I was on the right track, I got stuck in (in)possibilities of the tool and complexity of my design topic"
  - laws of Relation Algebra never applied by students

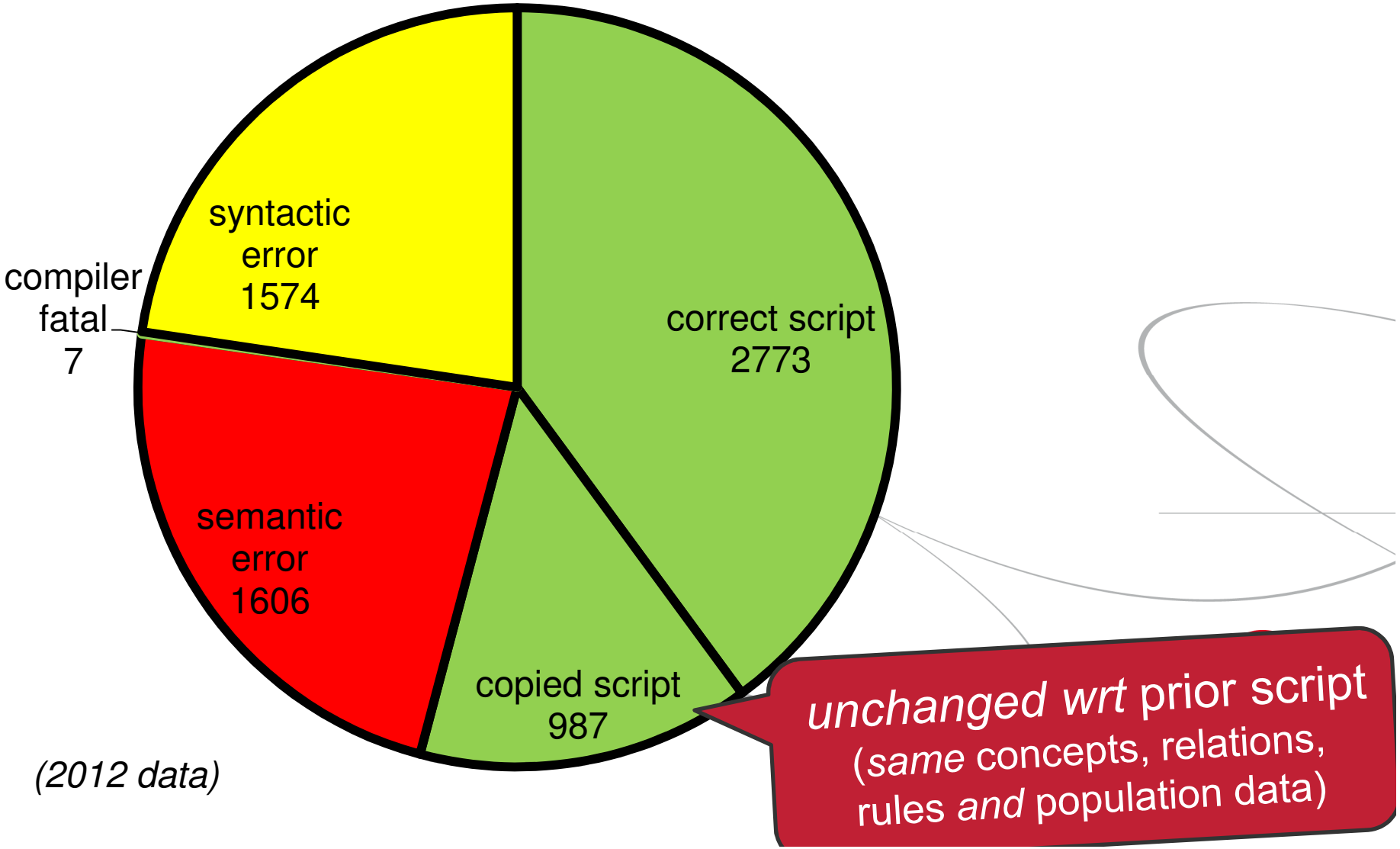
- **course materials: not OK**
  - "use of wiki + Ampersand tool is both hard and superfluous (should not be part of a course in Business Rules)"
  - "Proposition Logic is rather technical, and goes beyond the course objectives"
  - "I would prefer a course book in Dutch"
  - "overall, I expected better"

Student feedback: the ugly

- **not realistic**
  - "I do not think that the Ampersand method or tool are applicable in a real company such as mine"
  - "complexity of systems and information prohibits overview"
  - "what skills and competences are needed? Where and how to go forward in a real organisation to be effective?"

- **too much formalism, too little reality**
  - "the really interesting part is rule-elicitation using semi-formal language (RuleSpeak) but this is hardly addressed"
  - "it remains vague: how does Ampersand work in real life?"
  - "what does BRM look like in everyday practice?"

difficulty in writing correct code
(*from* Gerard Michels)



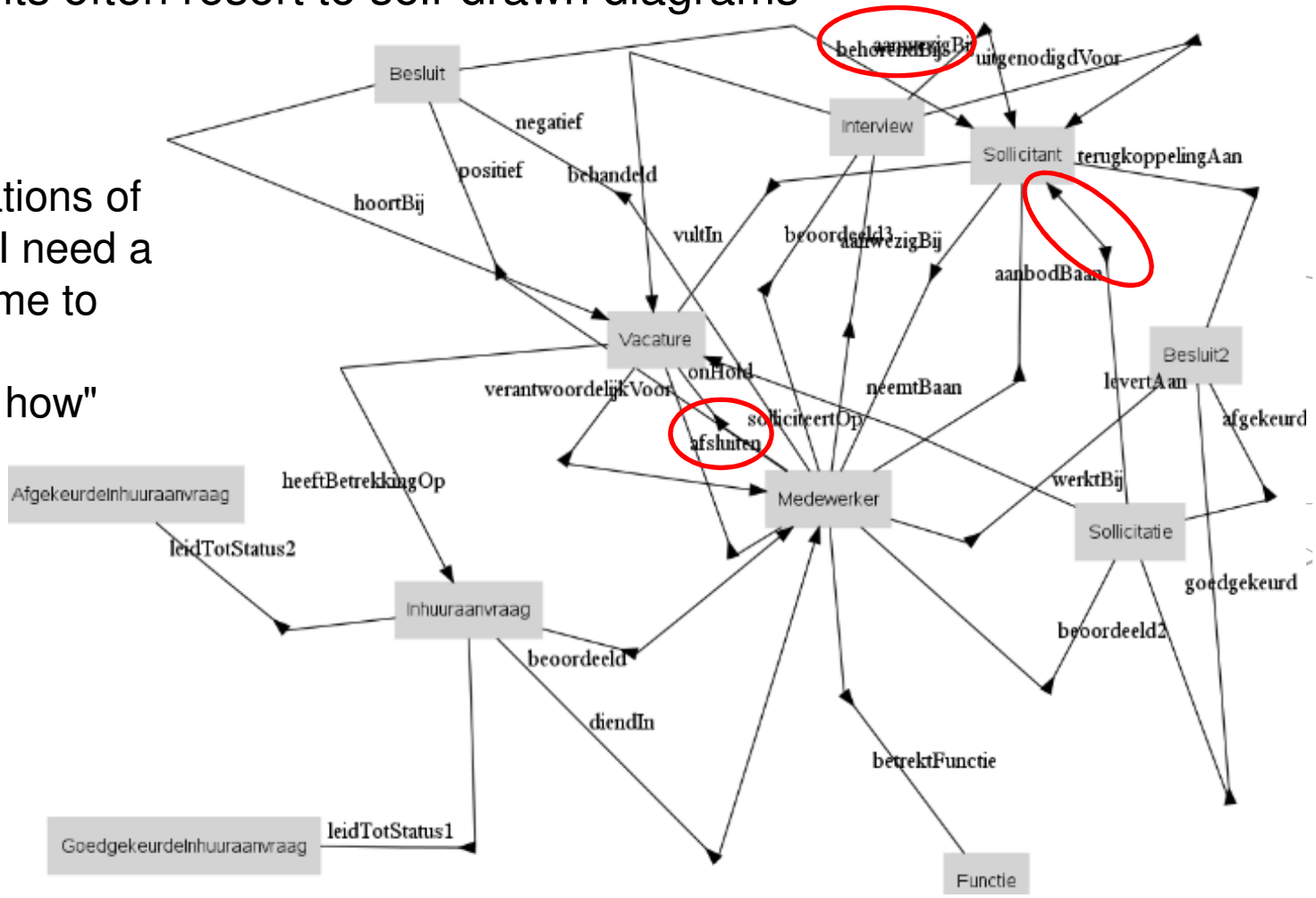
Student feedback on script language

- **difficulty in writing correct code**
  - variations in notations are confusing
  - why Explanation? Meaning? Pragma?
    - MEANING "An Undo-activity has 0 or 1 Confirmation. A Confirmation is related to 0 or 1 Undo-activity."
    - PRAGMA "Undo-activity has Confirmation"
  - unintelligible errors, e.g. for
    - ¨illegal¨ quotes
    - NAME ≠ Name
    - 'John' ≠ ' John'
    - entity-integrity / duplicate data
  - "I need good examples and design patterns for common solutions"

Student feedback on diagrams

- "Visualisation helps to obtain overview"
- ☹ students often resort to self-drawn diagrams

"to understand violations of a (compound) rule, I need a diagram that helps me to trace which tuples are composed, and how"



Student feedback on Ampersand tool

- **inadequate**
  - "time wasted in script testing, understanding the error-reports, and finding and fixing bugs"
  - "debugging is problematic: RAP2 reported a problem in line 92, but the error was in line 124"
  - "the operator -| produced false results"
  - "lack of good examples/patterns to illustrate the theory"
  - "to test the set of rules, I had to invent my own approach"
  - "when you edit data in several relations or rules at once, the analysis of violations becomes next to impossible"
  - "use of Ampersand is trial-and-error"
- **"why must we learn this particular tool?"**

design formalisms

- a good conceptual design sets the stage
- rules guide the business process
- translate back-&-forth natural language↔formal rules

design formalisms

- **BPMIT students are not good designers**
  - conceptual designs are often mediocre
  - student have difficulty even with multiplicity rules
  - compound rules are usually simple or wrong
- **their rules do not guide a business process**
  - rulesets are often ad-hoc, incomplete or incoherent
  - few students explain how to resolve a violation
  - even less try to explain how their rules guide the process
- **students are "lost in translation"**
  - focus is neither on rule elicitation nor on thorough rule-analysis
  - students sometimes verify a rule formula
  - students never validate in rea

Ampersand does not support both the rule owner **&** rule designer

how now with the BPMIT course

- **shift focus to Controlled Natural Language**
  - formalization is taken one step too far
  - textbook + tool do not fit the learning goals of BPMIT students
  - the logic should be there, the formalization should go under the hood
- **new course book**
  - should describe our BRA (its merits and rules) explicitly
  - should outline one suitable CNL (ours)
  - should cover rule elicitation (business-to-CNL) and validation (back)
  - should challenge students to validate the rules in real business
- **new Ampersand**
  - should be fed with Controlled Natural Language
  - should auto-generate realistic test data and violations
  - should provide GUI for tracing and editing